Agricultural Yield Estimation by Polynomial Regression

Amogh M K (01FB15EEC027)

Guide: Dr. V.K. Agrawal,
Department of Information Science and Engineering,
Crucible of Research and Innovation (CORI),
PES University, Bangalore-560085.

PES University,
Bangalore-560085.

Acknowledgements

On the submission of my project entitled **Agricultural Yield Estimation by Polynomial Regression**, I would like to express my gratitude to the project supervisor **Dr. V.K. Agrawal**, Director of CORI, PES University for his guidance and inspiration during the course of the project work.

Also, I would like to thank my family and friends for their support and encouragement.

Abstract

The main focus in this project was to gain experience in taking up and solving a regression problem.

This project aims to approximately predict the yield of a crop in a state through polynomial regression using Gradient descent algorithm and normal equation method in order to help the farmers, industries and other associated personnel for future planning.

Table of Contents

1 Introduction

With the impact of climate change in India, majority of the agricultural crops are being badly affected in terms of their performance over a period of last two decades. Predicting the crop yield well ahead of its harvest would help the policy makers and farmers for taking appropriate measures for marketing and storage. Such predictions will also help the associated industries for planning the logistics of their business.

Crop production is a complex phenomenon that is influenced by agro-climatic input parameters. Agriculture input parameters varies from field to field and farmer to farmer. Collecting such information on a larger area is a daunting task. However, the climatic information collected in India at every square meter area in different parts of the area tabulated by Indian Meteorological Department. Also, the yield of every crop in each state is collected and published by the department of agriculture and cooperation every year.

Such data sets are used in this project for predicting the influence on major crops and thus, their yield in a future year.

2 Concepts

2.1 Machine Learning

Machine learning is an emerging technology that can aid in the discovery of rules and patterns in sets of data. It has frequently been observed that the volume of recorded data is growing at an astonishing rate that far outstrips our ability to make sense of it.

Machine Learning has been defined in many ways:
Field of study that gives computers the ability to learn without being explicitly programmed.
- Arthur Samuels
A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance of T, as measured by P, improves with experience E.
- Tom Mitchell

Machine learning algorithms are typically classified into two broad categories, depending on the nature of the learning data available to a learning system. These are:
Supervised Learning: The computer is presented with example inputs and their desired outputs and the goal is to learn a general rule that maps inputs to outputs.
Unsupervised Learning: A computer program interacts with a dynamic environment in which it must perform a certain goal. The program is provided feedback in terms of rewards and punishments as it navigates its problem space.

Another categorization of machine learning tasks arises when one considers the desired output of a machine-learned system:
Classification: Inputs are divided into two or more classes, and the learner must produce a model that assigns unseen inputs to one or more of these classes. This is typically tackled in a supervised way.
Regression: Similar to classification but the outputs are continuous rather than discrete. This is also a supervised problem.

2.2 Polynomial Regression

Regression is a statistical method that allows us to summarize and study relationships between two continuous (quantitative) variables.

If the plot of the training data suggests that there is a linear relationship between the 2 variables, a linear regression model is constructed to fit a line to the set of points.

Sometimes, a plot of the residuals versus a predictor may suggest there is a nonlinear relationship. One way to try to account for such a relationship is through a polynomial regression model.

Such a model for a single predictor, X, is:

$$Y = \theta_0 + \theta_1 X + \theta_2 X^2 + ... + \theta_n X^n \tag{1}$$

n - degree of the polynomial.
θ - weights or parameters.

For lower degrees, the relationship has a specific name (i.e., h = 2 is called quadratic, h = 3 is called cubic, h = 4 is called quartic, and so on).

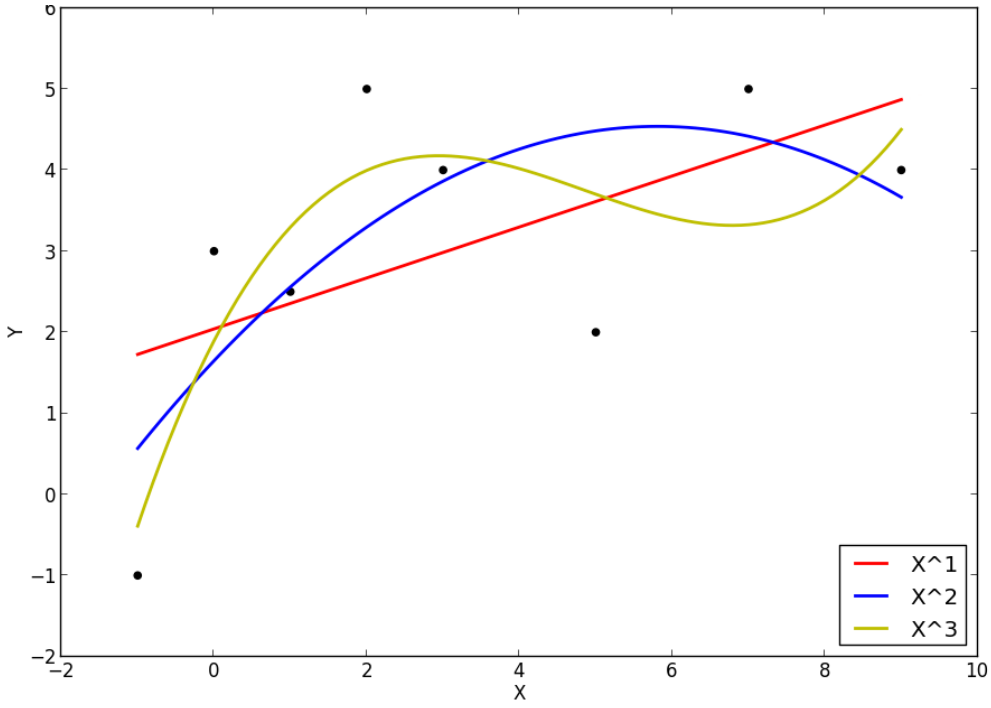


Fig. 1. Linear regression and 2nd, 3rd degree polynomial regression over the same data set

2.3 Cost function

Cost function is a function that maps an event or values of one or more variables onto a real number intuitively representing some "cost" associated with the event. An optimization problem seeks to minimize a cost function. Typically a cost function is used for parameter estimation, and the event in question is some function of the difference between estimated and true values for an instance of data.

Parameter estimation for supervised learning tasks such as regression or classification can be formulated as the minimization of a cost function over a training set in order to find a function that models its input well. The cost function quantifies the amount by which the prediction deviates from the actual values.One such cost function is “Squared error function” which is given below:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2 \tag{2}$$

Where, m is the no. of training examples.

The mean is halved as a convenience for the computation of the gradient descent, as the derivative term of the square function will cancel out the 1/2 term.
On minimizing this function, we obtain theta values which in turn gives us the function that models input well.

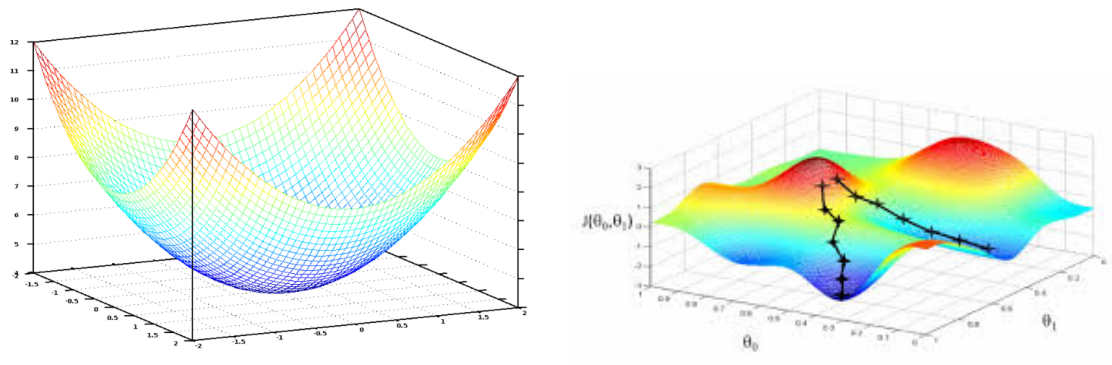


Fig. 2.
Different Cost Functions:
a) Bowl shaped cost function - only one minima.
b) Cost function with one global minima and multiple local minima.

2.4 Gradient Descent

Gradient descent is one of those "greatest hits" algorithms that can offer a new perspective for solving problems.

Gradient descent is an algorithm that minimizes functions. Given a function defined by a set of parameters, gradient descent starts with an initial set of parameter values and iteratively moves toward a set of parameter values that minimize the function. This iterative minimization is achieved using calculus, taking steps in the negative direction of the function gradient.

Gradient descent is the algorithm used here to effectively minimize the cost function.

The gradient descent algorithm is defined as:

Repeat until convergence:

$$\theta_j = \theta_j - \alpha \frac{\partial(J(\theta))}{\partial\theta} \tag{3}$$

At each iteration j, one should simultaneously update the parameters θ_1,θ_2,...,θ_n. Updating a specific parameter prior to calculating another one on the j^{th} iteration would yield to a wrong implementation.

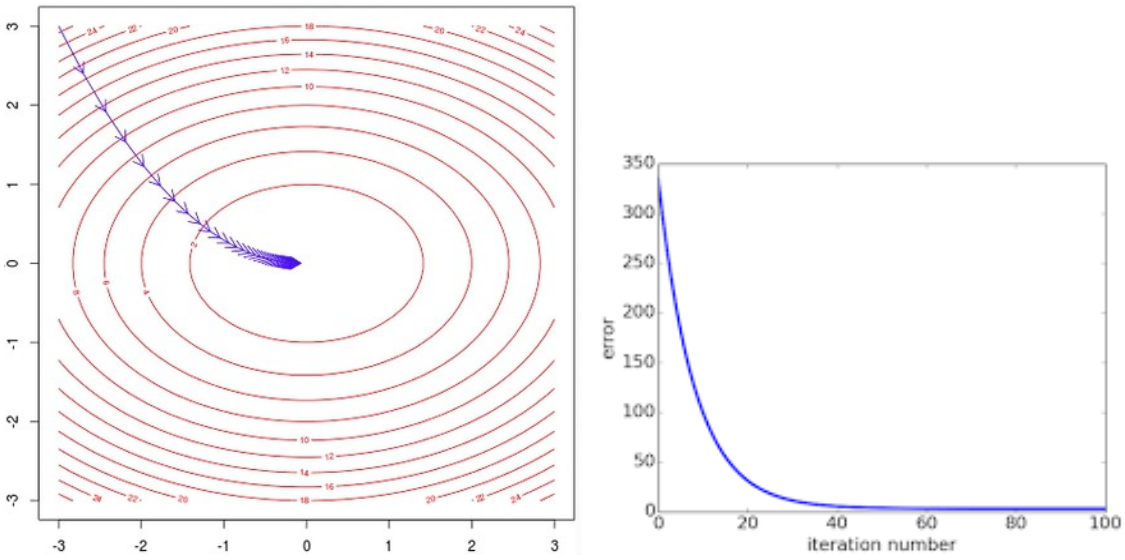


Fig. 3.
a) Working of Gradient Descent to obtain θ values corresponding to minima of cost function.
b)Decrease in cost value after each iteration of Gradient Descent Algorithm.

2.5 Normal Equation

Gradient descent gives one way of minimizing cost function. Normal Equation performs the minimization explicitly and without resorting to an iterative algorithm.

In this method, cost function is minimized by explicitly taking its derivatives with respect to the θ_j's, and setting them to zero. This allows us to find the optimum theta without iteration.

The normal equation formula is given below:

$$\theta = (X^T X)^{-1} X^T Y \tag{4}$$

There is no need to do feature scaling with the normal equation.
The following is a comparison of gradient descent and the normal equation:

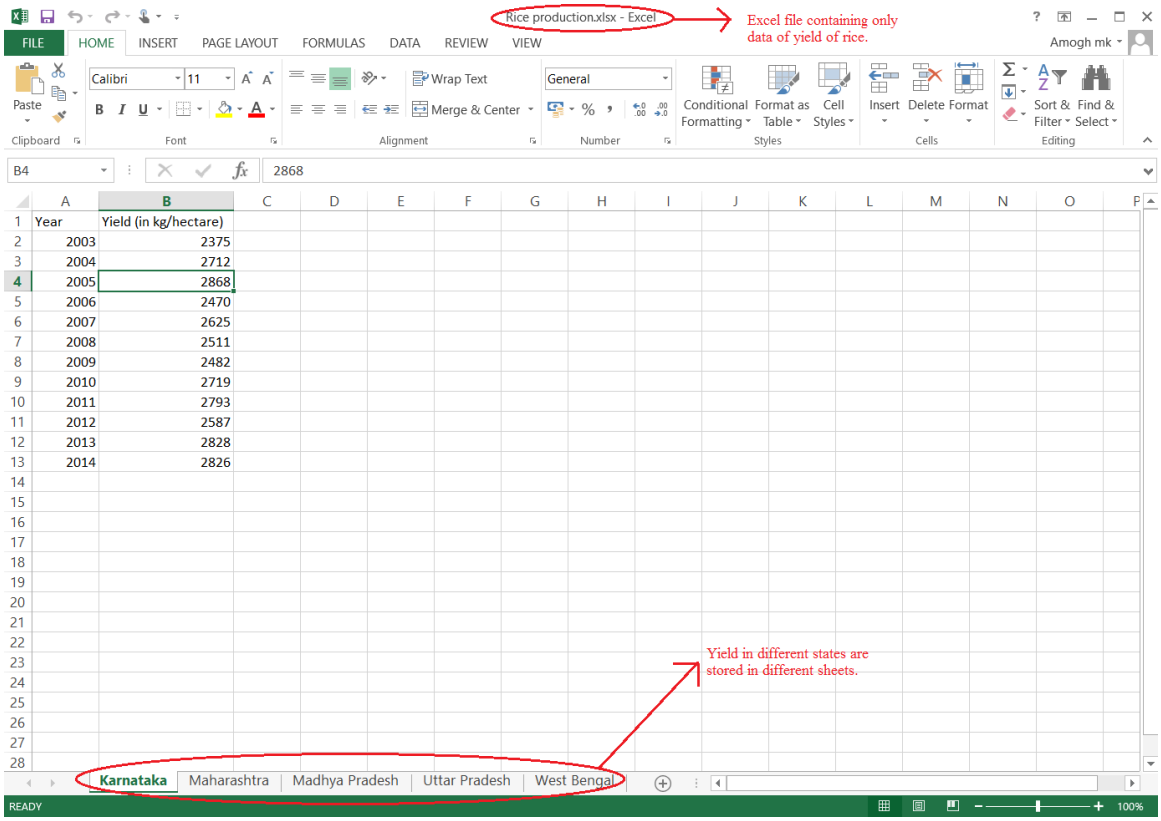
Gradient Descent	**Normal Equation**
Need to choose alpha.	No need to choose alpha.
Needs many iterations.	No need to iterate.
Time complexity is $O(kn^2)$.	Time complexity is $O(n^3)$.
Works well when n is large.	Slow if n is very large.

With the normal equation, computing the inversion has complexity O(n3). So if we have a very large number of features, the normal equation will be slow. In practice, when n exceeds 10,000 it might be a good time to go from a normal solution to an iterative process.

3 Overview of Data

The data used for the project are stored in separate excel files and has to be accessed and trained every time a prediction has to be done.

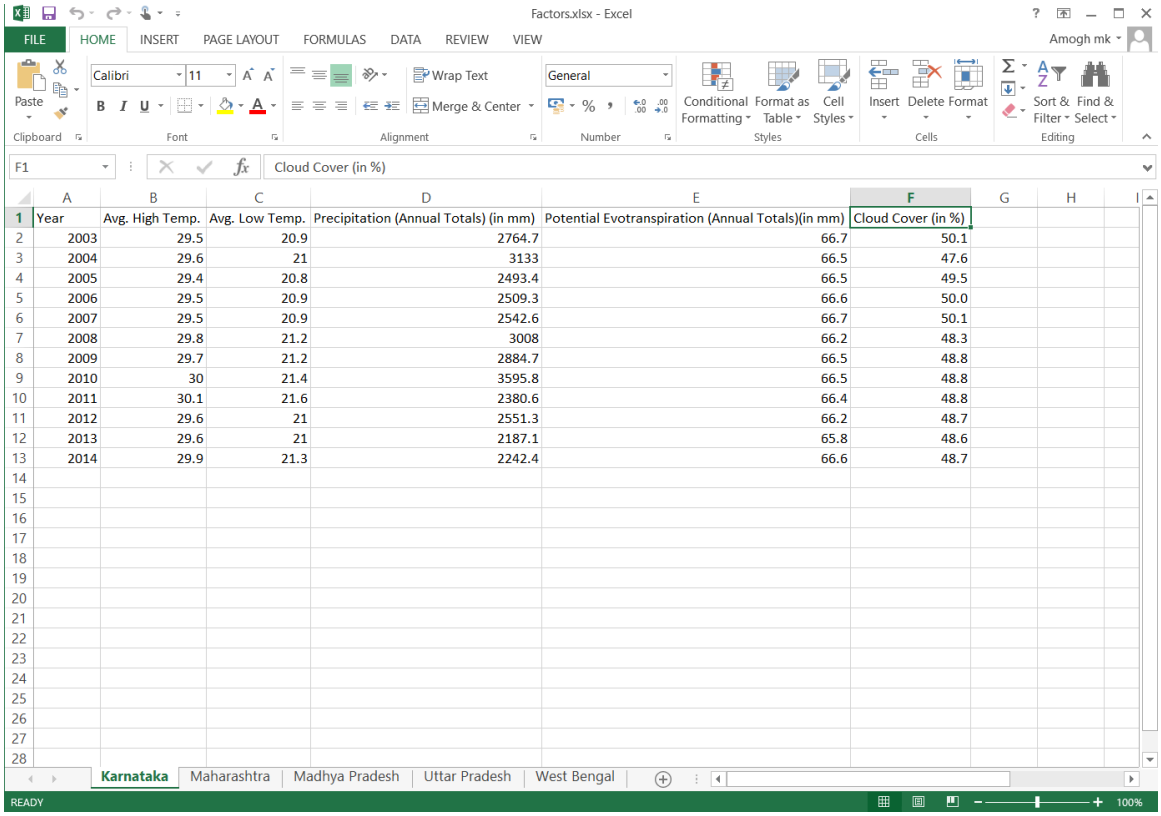
The yield of each crop is stored in a different excel file, with each file consisting of sheets containing data of yield of that crop in different states over a set of years.



Year	Yield (in kg/hectare)
2003	2375
2004	2712
2005	2868
2006	2470
2007	2625
2008	2511
2009	2482
2010	2719
2011	2793
2012	2587
2013	2828
2014	2826

Fig. 4.

Factors affecting the yield of a crop that are taken into consideration in this project are **Average low and high temperatures, Percentage of Cloud Cover, Precipitation and Potential Evaporation.** Values of temperature and cloud cover are averaged over the time period of a year in each state. However, data of precipitation and evaporation are stored as annual totals in each state. These data are stored in yet another excel file.



Factors.xlsx - Excel

Amogh mk

Cloud Cover (in %)

	A	B	C	D	E	F	G	H
1	Year	Avg. High Temp.	Avg. Low Temp.	Precipitation (Annual Totals) (in mm)	Potential Evotranspiration (Annual Totals)(in mm)	Cloud Cover (in %)		
2	2003	29.5	20.9	2764.7	66.7	50.1		
3	2004	29.6	21	3133	66.5	47.6		
4	2005	29.4	20.8	2493.4	66.5	49.5		
5	2006	29.5	20.9	2509.3	66.6	50.0		
6	2007	29.5	20.9	2542.6	66.7	50.1		
7	2008	29.8	21.2	3008	66.2	48.3		
8	2009	29.7	21.2	2884.7	66.5	48.8		
9	2010	30	21.4	3595.8	66.5	48.8		
10	2011	30.1	21.6	2380.6	66.4	48.8		
11	2012	29.6	21	2551.3	66.2	48.7		
12	2013	29.6	21	2187.1	65.8	48.6		
13	2014	29.9	21.3	2242.4	66.6	48.7		

Karnataka | Maharashtra | Madhya Pradesh | Uttar Pradesh | West Bengal

READY

Fig. 5.

Note:

The data used in this project are not very accurate as exact data was unavailable. Hence, the data has been tweaked a little in order to fit the needs of this project.

4 Working methodology

The main aim of this project is to predict the yield of a crop in a future year. When the user chooses the state and the crop, the yield is predicted through the following steps:

- At the beginning, the program accesses the data of the yield of that crop in that particular state over the past years and trains that data to predict what the yield would be based on the previous yields. Let this yield be *yieldpredict1*.
- The program then predicts the values of the factors affecting the yield of the crop in that particular year by accessing and training the data of factors in the state during the past few years.
- Furthermore, through the predicted values of the factors, the program predicts three more yield values *yieldpredict2*, *yieldpredict3* and

yieldpredict4 which are predicted through the predicted average high and low temperatures, precipitation and evaporation values and cloud cover percentage respectively.

– At the end, a final yield value is obtained which is the mean of all 4 predicted yield values (*yieldpredict1*, *yieldpredict2*, *yieldpredict3*, *yieldpredict4*).

5 Implementation

The project was entirely implemented using Python programming language. Also, a graphical user interface (GUI) was designed using the tkinter library of Python. Other libraries such as numpy, matplotlib, xlrd were also used for the implementation of polynomial regression, plotting graphs and extracting data from the excel files respectively.

```
def polynomial_regression(self, x, y, deg):
    self.order = deg+1
    features = np.empty([len(x), self.order], dtype=float)
    self.theta = np.zeros(self.order)
    cost_prev = 0

    for i in range(len(x)):
        for p in range(self.order):
            features[i][p] = pow(x[i], p)

    log = 0
    for repeat in range(self.max_iter):
        mat_mult = (features.dot(self.theta) - y)
        self.theta[0] -= (self.learning_rate/len(x))*(mat_mult*features[:,0]).sum()
        for j in range(1, self.order):
            self.theta[j] -= (self.learning_rate/len(x))*((mat_mult*features[:,j]).sum()
                                                          + self.l1_penalty + self.l2_penalty*self.theta[j])
        if(repeat%10 == 0):
            cost = (1./(2*len(x))) * (np.power(features.dot(self.theta) - y, 2).sum()
                                      + self.l1_penalty*self.theta.sum()
                                      + self.l2_penalty*np.power(self.theta,2).sum())
            if(log > 1 and np.abs(cost - cost_prev) < self.tolerance):
                break
            log += 1
            cost_prev = cost
```

Fig. 6. Implementation of Gradient Descent Algorithm in Python.

```
def NormalEquation(x_val,y,*x):
    m=len(y)
    y=(np.array([y])).transpose()
    X=np.ones((m,1))
    for i in x:
        temp=(np.array([i])).transpose()
        X = np.hstack([X,temp])
    n=(X.shape)[1]
    theta=np.zeros((n,1))
    theta=(np.linalg.inv(X.transpose().dot(X))).dot(X.transpose().dot(y))
    x_val2=np.empty((1,n))
    x_val2[0,0]=1
    for i in range(len(x_val)):
        x_val2[0,i+1]=x_val[i]
    y_val=x_val2.dot(theta)
    return(y_val[0,0])
```

Fig. 7. Implementation of Normal Equation method in Python.

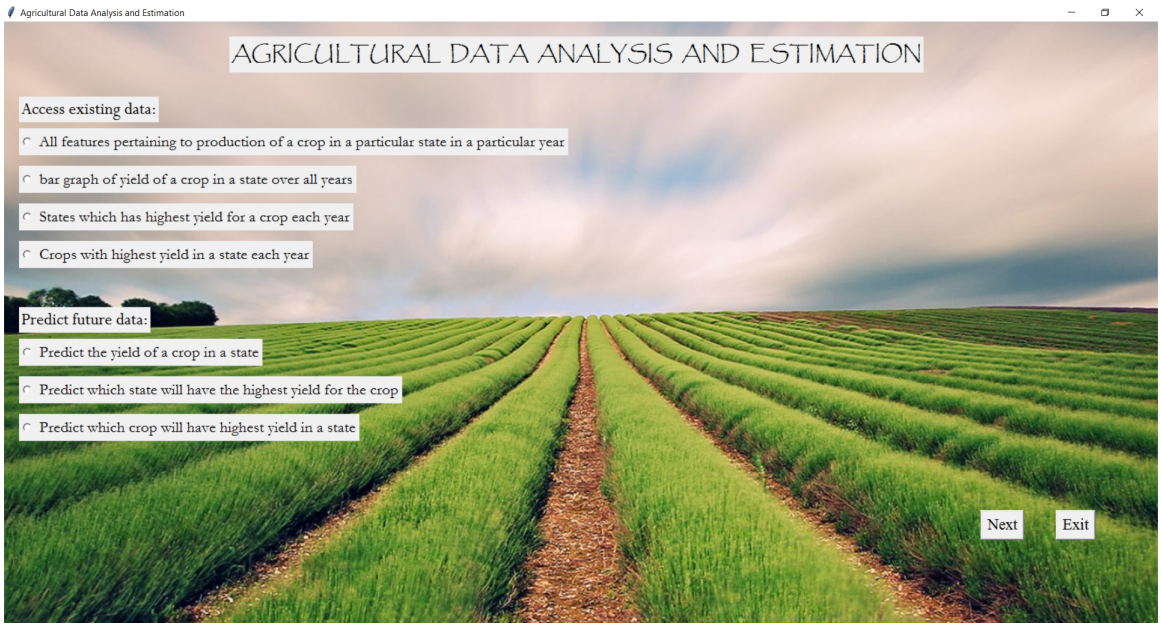


Fig. 8. Main window of GUI offering a list of Options.

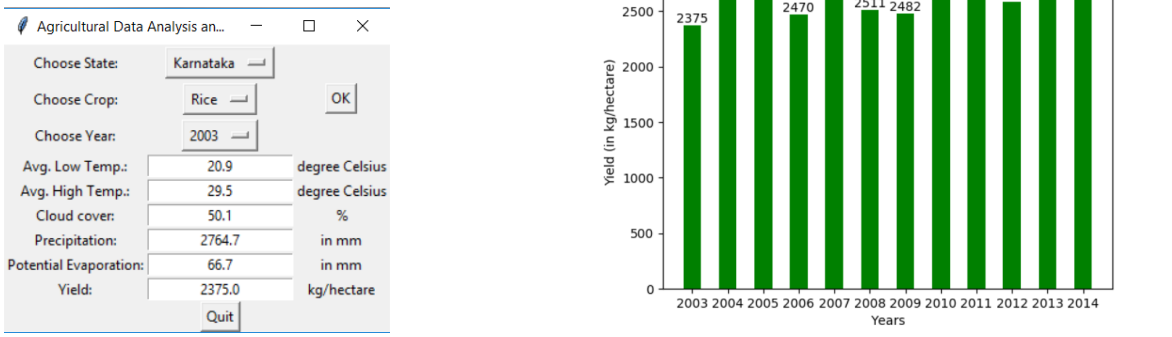


Fig. 9.
Choice 1: Access all data about the growth of a crop in a state in a particular year.
Choice 2: Access data of yield of a crop in a state over all the previous years.

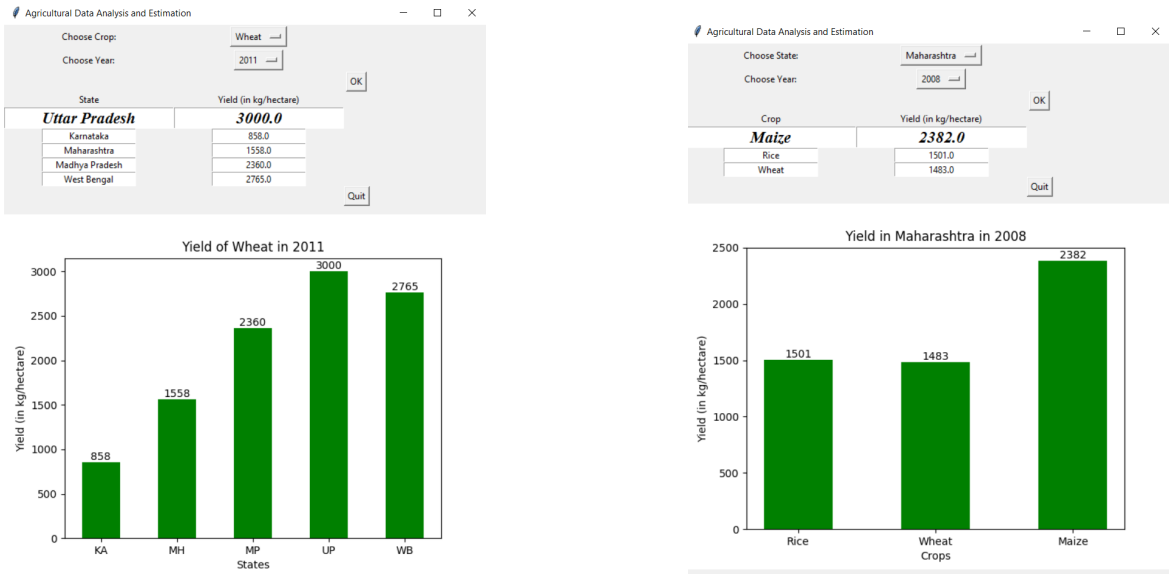


Fig. 10.
Choice 3: Access data of yield of a crop in all states in a year.
Choice 4: Access data of yield of all crops in a state in a year.

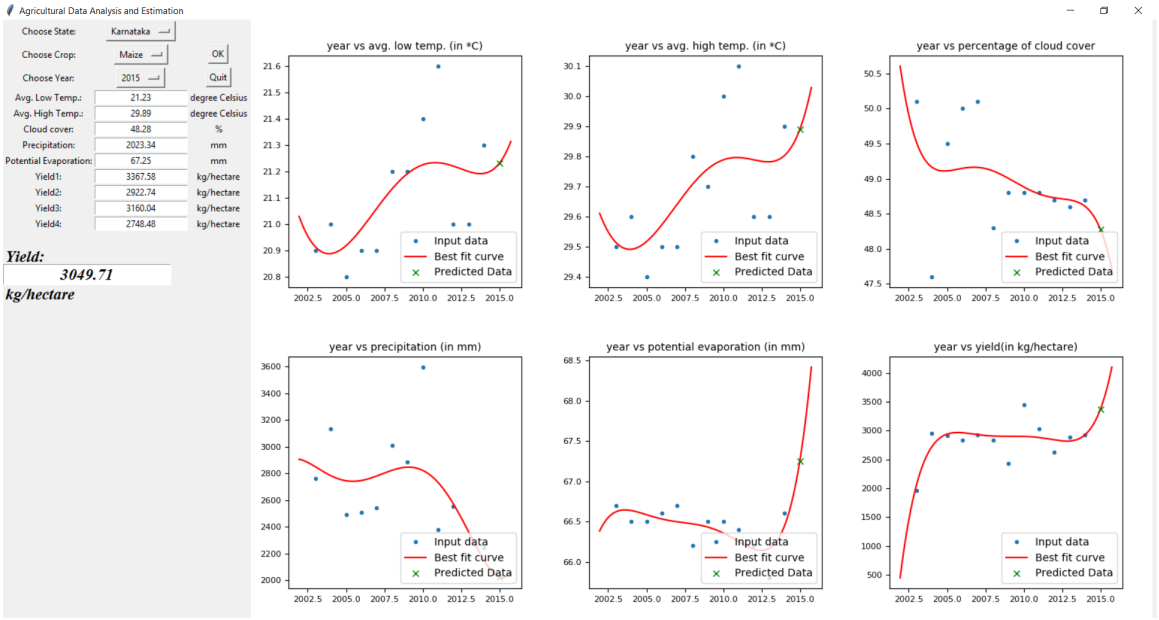
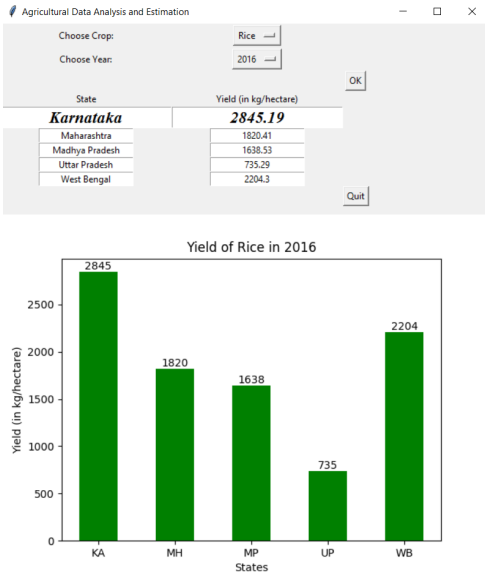


Fig. 11. Choice 5: Predicting the yield of a crop in a State.



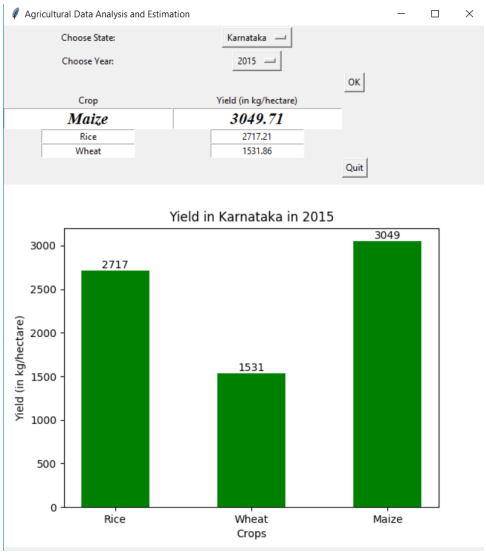


Fig. 12.
Choice 6: Predicting which state will have the maximum yield for a crop.
Choice 7: Predicting which crop will have maximum yield in a state.

6 Results

The approximate yield predicted in this project has varying accuracy depending on the crop whose yield is predicted and the state in which the crop will be grown.

For example:

- The yield of *Maize* in *Karnataka* in *2015* is predicted to be approximately **3050 kg/hectare**. The yield in the past 3 years were **2620, 2883, 2921 kg/hectare**. Hence, this prediction can be termed as being pretty accurate.
- However, the yield of *Maize* in *West Bengal* in *2015* is predicted to be approximately **3264 kg/hectare**. The yield in the past 3 years were **3947, 4059, 4347 kg/hectare**. This prediction is not accurate as we see a huge decline in yield. This is due to fact that other states have lesser yield of *Maize* even though their weather conditions vary only by a small amount and the project takes the values of factors in all the states into consideration while predicting the yield.
- Similarly, the yield of *Wheat* in *Karnataka* in *2015* is predicted to be approximately **1532 kg/hectare**. The yield in the past 3 years were **735, 1075, 1091 kg/hectare**. Here, we can observe considerable increase which is due to the fact that other states have higher yield of *Wheat* even though their weather conditions vary only by a small amount.

7 Conclusion and Future work

The main objective of this project was to approximately predict the yield in a future year. Even though this was achieved, there were considerable errors in the predictions as the values varied by a considerable amount in a few cases as the project considered only a few factors such as temperatures, cloud cover, precipitation and evaporation for the prediction. But the crop yield also depends on various other factors such soil porosity, topography, amount and type of fertilizers used and many more. Also, we have to take the fact into account that development and technological advancements in each state vary greatly as this hugely affects the prediction of yield.

Hence, we can improve this project by taking all these various factors into account and by implementing other complex regression algorithms which are more accurate.

Also, this project can be extended to predict the cost in which the crops will be sold depending on their need, imports, exports etc. and hence predict the amount of profits and losses. This will be especially useful for all farmers and related industries as it will help them to plan their business ahead of time.

References

- AGRICULTURAL STATISTICS AT A GLANCE, Ministry of Agriculture, Department of Agriculture and Cooperation, Directorate of Economics and Statistics, Government of India.
- http://www.indiawaterportal.org/met_data/
- http://eands.dacnet.nic.in/latest_20011.htm
- https://en.wikipedia.org/wiki/Machine_learning
- https://www.coursera.org/learn/machine-learning/home/welcome